In the Specification

Please amend the specification of this application as follows:

Rewrite the paragraph at page 1, lines 4 to 5 as follows:

--Serial Number 09/154,385 entitled "METHOD OF INITIALIZING A CPU CORE FOR EMULATION" filed September 16, 1998, now U.S. Patent No. 6,167,385 issued December 26, 2000; and--

Rewrite the paragraph at page 1, lines 7 to 9 as follows:

--Serial Number 09/483,367, entitled "EMULATION SUSPEND MODE WITH DIFFERING RESPONSE TO DIFFERING CLASSES OF INTERRUPTS" claiming priority from U.S. Provisional Application No. 60/120,809 filed February 19, 1999;--

Rewrite the paragraph at page 1, lines 10 to 11 as follows:

--Serial Number 09/481,852, entitled "EMULATION SUSPENSION MODE WITH STOP MODE EXTENSION" claiming priority from U.S. Provisional Application No. 60/120,809 filed February 19, 1999;--

Rewrite the paragraph at page 1, lines 12 to 13 as follows:

--Serial Number 09/483,568, entitled "EMULATION SUSPEND MODE HANDLING MULTIPLE STOPS AND STARTS" claiming priority from U.S. Provisional Application No. 60/120,809 filed February 19, 1999;--

Rewrite the paragraph at page 1, lines 14 to 15 as follows:

--Serial Number 09/483,697, entitled "EMULATION SUSPEND MODE WITH FRAME CONTROLLED RESOURCE ACCESS " claiming priority from U.S. Provisional Application No. 60/120,809 filed February 19, 1999;--

Rewrite the paragraph at page 1, lines 16 to 17 as follows:

--Serial Number 09/482,902, entitled "EMULATION SUSPEND MODE WITH INSTRUCTION JAMMING" claiming priority from U.S. Provisional Application No. 60/120,809 filed February 19, 1999;--

Rewrite the paragraph at page 1, lines 18 to 19 as follows:

--Serial Number 09/483,570, entitled "SOFTWARE EMULATION MONITOR EMPLOYED WITH HARDWARE SUSPEND MODE" claiming priority from U.S. Provisional Application No. 60/120,683 filed February 19, 1999;--

Rewrite the paragraph at page 1, lines 20 to 22 as follows:

--Serial Number 09/483,273, entitled "EMULATION SYSTEM WITH SEARCH AND IDENTIFICATION OF OPTIONAL EMULATION PERIPHERALS" claiming priority from U.S. Provisional Application No. 60/120,960 filed February 19, 1999;--

Rewrite the paragraph at page 1, lines 23 to 25 as follows:

--Serial Number 09/483,783, entitled "~~EMULATION~~ SYSTEM WITH ADDRESS COMPARISON UNIT AND ~~DATA~~ COMPARISON UNIT OWNERSHIP ARBITRATION" ~~claiming~~ priority from U.S. Provisional Application No. 60/120,791 filed February 19, 1999; and --

Rewrite the paragraph at page 1, lines 26 to 28 as follows:

--Serial Number 09/481,853, entitled "EMULATION SYSTEM WITH PERIPHERALS RECORDING EMULATION FRAME WHEN STOP GENERATED" claiming priority from U.S. Provisional Application No. 60/120,810 filed February 19, 1999.--

Rewrite the paragraph at page 3, line 24 to page 4, line 6 as follows:

--Another problem is product emulation when employing these programmable digital processors. Product development and debugging is best handled with an emulation circuit closely corresponding to the actual integrated circuit to be employed in the final product. In circuit emulation (ICE) is in response to this need. An integrated circuit with ICE includes auxiliary circuits not needed in the operating product included solely to enhance emulation

visibility. In the typical system level integration circuit, these emulation circuits use only a very small fraction of the number of transistors employed in operating circuits. Thus it is feasible to include ICE circuits in all integrated circuits manufactured. Since every integrated circuit can be used for emulation, inventory and manufacturing need not differ between a normal product and an emulation enhanced product.--

B2

Rewrite the paragraph at page 5, lines 2 to 18 as follows:

--This invention involves emulation communications via a test access port and boundary-scan architecture providing serial access to a serial connection of a plurality of registers disposed in a plurality of modules. One of the modules is selected for communication. Nonselected modules are made nonresponsive to data on the serial connection. The external emulation hardware supplies a serial signal having a first logic state for a number of cycles greater in number than a number of bits of the serial connection of registers to the test access port. The emulation hardware supplies a start bit having an opposite logic state. The selected module detects the start bit and stores the next predetermined number of data bits. These bits could be data bits to be stored in a program visible data register or bits interpreted as an instruction for execution by the module. The selected module may transmit return communications via the serial scan path using the same format.--

B3

Rewrite the paragraph at page 10, lines 6 to 17 as follows:

--The preferred embodiment of this invention includes an extension to the JTAG interface. Two pins nET0 and nET1 serve as a two pin trigger channel function. This function supplements the serial access capability of the standard interface with continuous monitoring of device activity. The two added pins create debug and test capabilities that cannot be created with the standard

B4

interface. The nET0 signal is called Emulation and Test 0 Not. This signal helps create a trigger to channel zero. Similarly, the nET1 signal is called Emulation and Test 1 Not. This signal helps create a trigger to channel one. These channels will be further explained below.--

Rewrite the paragraph at page 10, lines 18 to 26 as follows:

--Figure 3 illustrates an emulation level view of target system 3. Target system 3 may include plural devices 11, 13 and 15. Figure 3 illustrates details of example device 13 which includes plural megamodules 21, 23 and 25. Figure 3 illustrates details of example megamodule 23. Example megamodule 23 includes debug and test control unit 30 and plural device domains. These device domains include central processing unit (CPU) core 31, analysis unit 33, memory 35 and debug/test direct memory access (DT_DMA) unit 37.--

Rewrite the paragraph at page 11, lines 3 to 15 as follows:

--Figure 4 illustrates an electrical connection view of the coupling between access adapter 2 and target system 3. Figure 4 shows the connections of the of the various signals of the JTAG header 5 illustrated in Figure 2. All these signals are connected to scan controller 41. The signals nTRST, TCK and TMS are connected to two example megamodules 31 and 33. Figure 4 illustrates the optional connection of TCKO to the target system clock SYSCLK. The scan input TDI connects to a scan input of megamodule 31. The scan output of megamodule 31 supplies the scan input of megamodule 33. The scan output of meg module 33 supplies the scan output TDO. The two extension signals nET0 and nET1 control megamodules 31 and 33 via merge unit 32. These extension signals are monitored by test equipment 43.--

Rewrite the paragraph at page 11, lines 17 to 24 as follows:

--The debugging environment illustrated in Figures 1 to 4 permit the user to control application execution by any programmable digital processor of target system 3. Typical control processes include: keyboard directives such as run, halt and step; software breakpoints using op-code replacement; internal analysis breakpoints specified program counter or watchpoints specified by data accesses; and externally generated debug events.--

Rewrite the paragraph at page 11, line 24 to page 12, line 2 as follows:

--Actions such as decoding a software breakpoint instruction (DSTOP), the occurrence of an analysis breakpoint or watchpoint (ASTOP), or the occurrence of a debug host computer event (HSTOP) are referred to as debug events. Debug events cause execution to suspend. Debug events tied to the execution of specific instructions are called breakpoints. Debug events generated by memory references are called watchpoints. External debug events can also suspend execution. Debug events cause entry into the Debug State.--

Rewrite the paragraph at page 12, lines 8 to 16 as follows:

--Execute state 101 corresponds to the ordinary operation of target device 3. In the execute state 101 instructions are executed by the programmable digital processor in normal fashion. There are no outstanding debug suspend conditions. A low logic level applied to the nTRST terminal or a software directive requesting functional run forces the operational state to execute state 101. In execute state 101 the pipeline fetches and executes instructions and processes interrupts in a normal way.--

Rewrite the paragraph at page 16, line 31 to page 17, line 11 as follows:

--The example system in Figure 4 shows the system connectivity necessary for debug with one or more devices, one containing a programmable digital processor core. Figure 4 omits signal buffering and other electrical considerations necessary to create a functional system. In this example, target device 3 contains two megamodules 31 and 33. Megamodule 31 includes a programmable digital processor core while megamodule 33 does not. The two devices share a parallel connection to signals nTRST, TCK, and TMS. The scan path begins as TDI at the connector, enters megamodule 31, exits megamodule 33, and ends as TDO back at the connector. Connections between merge unit 32 and pins nET1 and nET0 create trigger channels one and zero.--

Rewrite the paragraph at page 18, lines 17 to 30 as follows:

--This invention proposes employing another data transfer protocol on the serial scan path. Figure 7 illustrates this alternate data transfer protocol. This data transfer protocol includes a first section 121 of plural bits of the same digital state. A second section 130 includes a start bit 131 of the opposite digital state and a predetermined number of data bits 133. Lastly, there is a third section 140 of the first digital state. In the preferred embodiment the first digital state is all 1's and the start bit is a 0. All the modules except the module including the desired register are made insensitive to the scan data stream. The selected module receives the data stream and searches for the start bit. Upon detection of the start bit, the predetermined number of bits is captured.--